Energy-aware QoE and Backhaul Traffic Optimization in Green Edge Adaptive Mobile Video Streaming

Abbas Mehrabi, *Member, IEEE,* Matti Siekkinen, *Member, IEEE,* and Antti Ylä-Jääski, *Member, IEEE*

***Abstract*—Collaborative caching and processing at the network edges through mobile edge computing (MEC) helps to improve the quality of experience (QoE) of mobile clients and alleviate significant traffic on backhaul network. Due to the challenges posed by current grid powered MEC systems, the integration of time-varying renewable energy into the MEC known as green MEC (GMEC) is a viable emerging solution. In this paper, we investigate the enabling of GMEC on joint optimization of QoE of the mobile clients and backhaul traffic in particularly dynamic adaptive video streaming over HTTP (DASH) scenarios. Due to intractability, we design a greedy-based algorithm with self-tuning parameterization mechanism to solve the formulated problem. Simulation results reveal that GMEC-enabled DASH system indeed helps not only to decrease grid power consumption but also significantly reduce backhaul traffic and improve average video bitrate of the clients. We also find out a threshold on the capacity of energy storage of edge servers after which the average video bitrate and backhaul traffic reaches a stable point. Our results can be used as some guidelines for mobile network operators (MNOs) to judge the effectiveness of GMEC for adaptive video streaming in next generation of mobile networks.**

***Index Terms*—Green mobile edge computing (GMEC), DASH, Quality of experience (QoE), Fairness, Greedy-based algorithm.**

I. Introduction

Due to the prominent role of mobile video streaming over the Internet which according to [1] will account for more than about 63% of Internet traffic by 2021, significant research attentions has been paid to devising solutions for improving the quality of experience (QoE) of the clients. Noticeable characteristic of the research efforts is envisioning dynamic adaptive video streaming over HTTP (DASH) solutions which provide means that allow the media players to adopt to the varying network conditions by dynamically choosing the most sustainable video bitrate [2], [3], [4]. Network-assisted DASH solutions along with the evolutions in mobile edge computing (MEC) paradigm [5], software defined networking (SDN) and network function virtualization (NFV) have significantly contributed to the improvement of QoE of the end users. Content caching and retrieval at the edges of the network has been shown to reduce significantly the traffic burden on the backhaul of the network [6]. Joint optimization solutions have been also suggested in which the edge caching is jointly utilized with the processing capability of the edge servers resulting in noticeable reduction in video delivery latency [7]. The trade-off between QoE of mobile clients and created traffic on backhaul network in collaborative edge caching adaptive mobile video streaming has been addressed in [8].

Despite significant research works on edge computing and caching, the energy efficiency aspect of MEC has been paid less attention. Powering the MEC facilities using renewable energy, known as green MEC (GMEC), has attracted research attentions from both academia and industry due to its very low operational costs for mobile network operators (MNOs) [9], [10]. The existing studies on renewable energy-powered MEC systems in the literature mainly focus on designing solutions to improve the quality of service (QoS) parameters such as computational latency. Although the potential of energy harvesting intuitively results in reducing the grid energy consumption, the controlled integration of renewable energy into the grid power for handling the processing tasks in particularly DASH video streaming at the edges is a challenging task. In other words, designing MEC system by integrating the renewable energy with the objective of jointly optimizing the trade-off between the QoE of mobile video streaming clients and created backhaul traffic has been overlooked. Our main contributions in this work are summarized as follows:

- We design the GMEC-enabled system which aims to quantify the impact of integrating renewable energy into the MEC in particularly dynamic adaptive video streaming over HTTP (DASH) scenarios.
- We formulate the joint optimization of QoE of the mobile cients and the backhaul traffic in GMEC-enabled DASH system as an integer non-linear programming (INLP) problem and design a suboptimal algorithm using self-tuning parametrization mechanism to solve it.
- A proactive edge caching heuristic is also designed which utilizes the statistical information about clients retention with respect to different videos. Results of our performance evaluations which show the superiority of GMEC-enabled system can indeed act as guidelines for system designers to judge the effectiveness of GMEC for DASH video streaming in next generation of mobile networks.

The remainder of the paper is organized as follows: Related work is discussed in Section II and the proposed system architecture is detailed in Section III. The optimization problem is formulated in Section IV and the proposed algorithm is presented in Section V. Simulation results are discussed in Section VI and finally, Section VII concludes the paper.

Authors are with the Department of Computer Science, Aalto University, Espoo 02150, Finland.

Email:{abbas.mehrabidavoodabadi,matti.siekkinen,antti.yla-jasski}@aalto.fi

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2019.2918847, IEEE Transactions on Green Communications and Networking

II. Related Work

Due to the users mobility and time-varying wireless channel conditions, dynamic adaptive video streaming over HTTP (DASH) is the prominent standard used in nowadays video streaming systems [2]. DASH-based video streaming solutions provide the mechanisms which allow the streaming media to adapt dynamically to the most sustainable video bitrate based on its instantaneous throughput. Kua *et al.* provide a survey on rate-based adaptation techniques in DASH video streaming [11]. Client-based solutions may fail under the scenarios when multiple video streaming clients simultaneously compete over the shared wireless resources. Network assisted adaptation solutions have been proposed which facilitate the cooperation among the network elements toward optimal/fair bitrate allocation among the competing clients [12].

Toward satisfying the requirements of 5G networks, mobile edge computing (MEC) concept has been proposed by the European telecommunications standard institute (ETSI) which enables moving the contents to the edges of the network nearby the end users [5]. Authors in [13] proposed the edge computing assisted system for DASH video streaming with the objective of jointly maximizing the QoE of the clients, fair bitrate allocation and balancing the utilized resources among multiple base stations. Along with MEC, the video content caching and retrieval at the network edge within the radio access network (RAN) has been shown to be a promising solution to alleviate significantly the traffic burden on the backhaul network [6], [14], [15], [16]. However, the bitrate adaptation and edge caching solutions in these works do not take into account the optimization of QoE jointly with the traffic on the backhaul network. Similarly, Tran *et al.* utilize the processing capability of edge servers jointly with the edge caching which leads to further improvement in the performance of video streaming system [7], [17]. However, the focus of these works is on designing optimization solutions for reducing the video delivery latency without taking into account the parameters which impact the QoE of the clients.

From the energy consumption point of view, the mobile network infrastructures contribute to about 2% of total CO_2 emissions worldwide which will grow as the number of base stations increases [18]. Despite of significant grid energy saving using MEC, the processing and content caching at the mobile edges still causes noticeable energy consumption especially in areas with dense deployment [19]. The integration of ambient renewable energy into the MEC has attracted the attentions from both academia and industry [9], [10], [20], [21]. In renewable energy-powered systems, the design objective should be improving the system performance subject to the available harvested energy since the renewable energy comes with very low cost. However, the time-varying characteristics of environmental energy sources during different time intervals makes the scheduling mechanisms more complicated for renewable energy-powered MEC systems [19].

Xu *et al.* have investigated the problem of determining the optimal workloads of the edge servers and their processing speeds taking into account both the network condition and the energy side information (ESI) [10]. Mao *et al.* addressed the problem of optimal task offloading from the users to the edge servers considering the energy harvesting mobile devices using both the channel-state information (CSI) and ESI [9]. However, these two works consider only either one edge server or one mobile device which can not be extended to large scale system in which the spatial diversity in the energy harvesting at different regions should be taken into account. Discussions on optimal battery storage size and efficient use of harvested energy in green cellular networks are detailed in respectively [20] and [21]. The existing works mainly focus on the computing performance of renewable energy-powered MEC system.

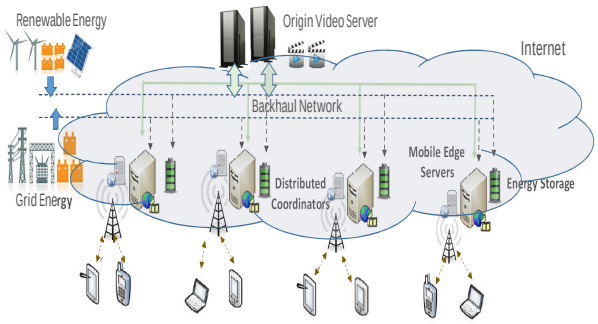


Fig. 1: Green mobile edge computing (GMEC) enabled DASH.

Different from the existing works, our objective in this paper is to investigate the impact of integrating renewable energy on the joint optimization of QoE of the mobile clients and the backhaul traffic for particularly DASH video streaming in MEC environments. To this end, we propose the joint QoE and backhaul traffic optimization in GMEC-enabled collaborative caching and processing considering the time-varying energy harvesting constraints. We then design an efficient energy-aware bitrate scheduling algorithm to solve the formulated optimization problem. The results of our performance evaluations reveal that GMEC-enabled DASH system indeed helps not only to decrease the grid power consumption of the edge servers but also significantly reduce the traffic on the backhaul network and further improve the average QoE of the clients. We further find out a threshold on the energy storage capacity of the servers such that the average achievable video bitrate and the backhaul traffic becomes stable after the threshold.

III. GMEC-enabled DASH

In this section, we first describe the architecture of the proposed system and then the notations.

A. System Overview

Fig. 1 represents the framework of the proposed green mobile edge computing (GMEC) enabled system for DASH video streaming. At the network edges, the mobile servers are associated with base stations (eNodeBs) from where the downlink resource blocks are allocated to the connected mobile clients according to proportional fairness (PF) policy. The majority of Internet traffic is from the video streaming services and it is also anticipated to be dominant in next generation of mobile networks according to the Cisco forecast [1]. Motivated by this fact, we do not consider the traffic

from other background mobile applications in our system model. Mobile edge servers cache locally the most frequently requested chunks/bitrates of the videos. Edge servers have also the processing capability such that the requested chunks with lower bitrates can be transrated from the same chunks but with higher bitrates which are available in local caches. In a hierarchical caching structure, the allocated chunk/bitrate to the client is directly downloaded from origin server in the cloud if there is no possibility of retrieval or processing at the local or neighborhood edge servers. Downloading the video chunks from the origin server causes outbound traffic on the core network which we refer to it as backhaul traffic.

In our system, the scheduling of DASH clients is performed within consecutive rounds, where each round consists of multiple discrete time slots with equal duration. We focus on only one round scheduling in this work. Within a scheduling round and in a collaborative manner, the distributed coordinators receive the clients radio access link level information from the base stations and the clients data (arrival/departure, buffer status) from their application software. Our system requires the explicit support from the clients application software such that the distributed coordinators can obtain the required information of the mobile clients. This communication design can be implemented through message-passing mechanism in order to obtain the network-assisted bitrate adaptation solution. After receiving the information, the coordinators perform two operations: first, the clients to edge servers mapping for which it is assumed that at each time slot, the client is assigned to nearest base station from where it receives the highest instantaneous signal-to-noise ratio (SNR) i.e., the ratio between the power of received signal from the base station to the power of background noise. Second, they solve the joint QoE and backhaul traffic optimization problem to determine the optimal and fair bitrate for each client. The optimization results are then explicitly communicated to the clients.

In addition to the limited amount of grid energy in each scheduling round, the edge servers can further utilize the renewable energy from the environment such as solar radiations in order to reduce grid energy consumption and improve processing efficiency [10]. As shown in Fig. 1, the edge servers are equipped with the energy storage which keep the harvested energy during different time periods of the day. Energy storages in our system are kind of lithium batteries which can be charged frequently even from the non-empty state. Corresponding to the volume of processed data, the edge servers consume an amount of energy. Each time the processing operation takes place at edge, the server consumes the storage energy first, if its available amount suffices for that processing, otherwise, the energy from the grid is consumed. The client's request can not be processed at the edge if none of the grid or the storage have sufficient power.

In large scale system deployment, the management of powering the edge servers from the renewable energy is performed by the local aggregators within different localities which in turn helps to reduce significantly the computing costs for the MNO. Furthermore, the installation of low-cost storages which is normally performed once per multiple years [22] brings significant grid energy saving throughout the year. Although there are associated maintenance costs with the energy storages which are relatively low during long term system operation [22], the noticeable improvement in the processing efficiency of the edge servers by integrating the renewable energy subsequently boosts the QoE of the mobile video streaming subscribers and therefore brings significant revenue for the MNOs.

B. System Notation

We consider one round scheduling of S number of DASH mobile clients which consists of $|T|$ discrete time slot each with fixed duration of Δt seconds. Multiple videos with different popularities are divided into the consecutive chunks each with fixed size of C seconds which are available in $|R|$ different resolutions initially stored at the origin server. K mobile edge servers are deployed in the system such that $1 \leq k \leq K$ refers to the server index throughout the paper. Also, the available downlink resource blocks at the base station associated with any edge server k, where $1 \leq k \leq K$, at time slot t is denoted by $W_k^{(t)}$. The available resource blocks at each time slot in any base station indicates the allocated bandwidth in the frequency domain based on the achievable throughput of the client and its assigned bitrate according to LTE 3GPP specifications [23], [25], [24]. For the sake of low complexity in the performance evaluation and following relevant research works [13], [26], the resource allocation to the clients at the base station is performed at every one-second time slot in our system model. However, our system is easily adoptable to smaller time scaling such as subframe without any modification to the model and the proposed solution. Arrival and departure time slots of client $1 \leq s \leq S$ i.e., the time slot when client s starts its video streaming session and the time slot that client either departs from the session or abandons its streaming, are represented by respectively A_s and D_s.

Binary variable $a_{sk}^{(t)}$ indicates the mapping of client s to edge server k at time slot t. The receivable downlink SNR and the theoretical throughput of client s from the associated base station (edge server) k at time slot t are denoted by respectively $SNR_{sk}^{(t)}$ and $Thr_{sk}^{(t)}$. Also, the achievable throughput of the client which is computed based on its theoretical throughput and the number of active clients connected to the same base station at time slot t, is also represented by $\hat{Thr}_{sk}^{(t)}$. Furthermore, the integer decision variable $r_{sk}^{(t)}$ indicates the video bitrate allocated to client s at edge server k by the coordinator in time slot t. Binary decision variables $x_{skc}^{(t)}$ and $x_{ske}^{(t)}$ are also defined which indicate that the requested chunk of client s assigned to server k at time slot t is downloaded from respectively the origin and the edge server. Binary decision variable $y_{sk}^{(t)}$ is defined such that $y_{sk}^{(t)} = 1$ indicates that the allocated chunk/bitrate to client s at time slot t is transrated at edge server k. Variable $tr_{sk}^{(t)}$ represents the bitrate (the same chunk available at edge server k) from which the bitrate of client s is transrated at time slot t and the constant factor ϕ represents the processing weight at each edge server. We have summarized the list of major system parameters including edge cache size and related parameters on energy harvesting/consumption at the edge in Table I.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2019.2918847, IEEE Transactions on Green Communications and Networking

TABLE I: Description of major system notations.

Notation	Description
K, S, R	Number of edge servers, number of DASH clients and the discrete set of available bitrates
$\|T\|$, Δt, C	Total number of scheduling time slots, the duration of each slot and the constant size of each video chunk in seconds
$W_k^{(t)}$	Available downlink resource blocks at base station k in time slot t
Q_M	Cache size at each edge server
$M_k^{(t)}$	Set of available chunks/bitrates in the cache of edge server k at time slot t
A_s, D_s	Arrival and departure times of client s
L_s^{max}, $L_s^{(t)}$	Maximum buffer capacity of client s and its buffer level at time slot t
BT_s	The backhaul data traffic caused by the video streaming of client s
$b_k^{(t)}$	Energy level of storage edge server k at the beginning of time slot t
$c_k^{(t)}$, e_p	Consumed energy from the storage of edge server k at time slot t and, the amount of consumed energy for processing per bit.
$c_k^{(t)}$, e_p	Consumed energy from the storage of edge server k at time slot t and, the amount of consumed energy for processing per bit
$h_k^{(t)}$, h_{max}	Harvested energy by edge server k at the beginning of active time slot t and, the maximum feasible amount of energy harvesting at each active time slot
B_e	Fixed energy storage capacity of each edge server
μ, σ^2	Mean time slot and the energy harvesting variance of the Gaussian-shaped function
ρ, ω, γ	Adjustable weighting parameters for average bitrate, bitrate switching and fairness, respectively
$a_{sk}^{(t)}$, $y_{sk}^{(t)}$	Binary indicator of allocating client s to server k at time slot t and, the binary decision variable indicating the processing of chunk/bitrate of client s at edge sever k in time slot t
$x_{ske}^{(t)}$, $x_{skc}^{(t)}$	Binary decision variables indicating the chunk download from respectively the edge and the origin server by client s allocated to edge server k at time slot t
$r_{sk}^{(t)}, tr_{sk}^{(t)} \in R$	Discrete video bitrate (integer decision variable) of client s at edge server k and, the transrated bitrate for client s at edge server k in time slot t

C. Edge Server Energy Model

Edge servers harvest the renewable energy during different time intervals from the solar panels which are installed in the deployment site. We consider the time-varying solar energy harvesting during one day time duration by approximating the amount of periodical harvested solar energy pattern presented in [27] using the Gaussian-shape function. In other words, the harvested energy by edge server k at the beginning of time slot t is given using the following function:

$$h_k^{(t)} \sim h_{max}\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\left(\frac{t-\mu}{\sigma}\right)^2}\right) \quad (1)$$

where h_{max} is the maximum feasible harvested energy at each time slot and μ and σ^2 are respectively the mean time slot and the variance of energy harvesting.

The amount of consumed energy by the edge server k at given time slot $1 \leq t \leq |T|$ depends on the volume of processed data at server k for all the active clients:

$$c_k^{(t)} = \sum_{s \in S} \sum_{1 \leq k' \leq K} a_{sk'}^{(t)} \cdot y_{sk}^{(t)} \cdot (tr_{sk}^{(t)} - r_{sk'}^{(t)}) \cdot \phi \cdot e_p \quad (2)$$

where index k' in the inner summation indicates a local or neighborhood edge server to which there are some connected clients at time slot t whose their requested chunk/bitrate is transrated at edge server k. Following linear energy evolution model of lithium battery [9], the available energy of storage of edge server k at the beginning of time slot t is given by:

$$b_k^{(t)} = min\{b_k^{(t-1)} - c_k^{(t-1)} + h_k^{(t)}, B_e\}, \ \ \forall 1 \leq t \leq |T| \quad (3)$$

Obviously, the overall consumed energy by the server for processing the clients' requested bitrates at each time slot t should be less than its available energy at that time slot.

$$c_k^{(t)} \leq b_k^{(t)}, \ \ \forall 1 \leq t \leq |T| \quad (4)$$

D. Quality of Experience and Fairness

As pointed out in several research studies, playback stalling caused by buffer underrun is the most critical influencing factor of QoE in video streaming. Therefore, we design the bitrate selection in such a way that stalling is completely avoided whenever possible, i.e., it is a constraint of the optimization problem. Other major factors in adaptive streaming are the perceived *average video bitrate* and the frequency and magnitude of *bitrate switching*.

1) Average Bitrate: As long as the video chunks are downloaded with higher bitrates, the client perceives the higher watching quality. Knowing the arrival and departure time slots, the average video bitrate perceived by client i i.e. the average of the bitrates allocated to the client by the coordinators during its streaming session is given by:

$$AQ_s = \frac{1}{|D_s - A_s|} \sum_{t=A_s}^{D_s} \sum_{k=1}^{K} a_{sk}^{(t)} \cdot r_{sk}^{(t)} \quad (5)$$

2) Bitrate Switching: The frequency of switching refers to the number of times that the bitrates of the consecutive chunks change and the switching magnitude is the amount of change in the bitrates of the consecutive chunks. The accumulated switching magnitude during the video streaming session of client i is given by:

$$E_s = \sum_{p=1}^{\lceil (D_s - A_s)/C \rceil} \sum_{k=1}^{K} (a_{sk}^{((p-1)C+1)} \cdot r_{sk}^{(p-1)C+1} - a_{sk}^{(p-2)C+1} \cdot r_{sk}^{(p-2)C+1}) \quad (6)$$

3) Fairness: Our system design takes explicit measures to ensure fairness when allocating the bitrates to the set of competing clients at each time slot. More precisely, the bitrates are allocated such that for every active client at each time slot, the difference between its allocated bitrate with the average bitrate of other simultaneous clients is minimized. We define F_s as the fairness value associated with the whole video streaming session of client s:

$$F_s = \sum_{t=A_s}^{D_s} \sum_{k=1}^{K} a_{sk}^{(t)} \cdot |r_{sk}^{(t)} - \bar{r}^{(t)}| \quad (7)$$

where $\bar{r}^{(t)}$ is the average bitrate of other simultaneous clients at time slot t.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2019.2918847, IEEE Transactions on Green Communications and Networking

E. Backhaul Data Traffic

With the decision variables defined in the system model, the overall backhaul data traffic during the whole video streaming duration of client s is given by the following relations:

$$BT_s = \sum_{t=A_s}^{D_s} \sum_{k=1}^{K} a_{sk}^{(t)} \cdot x_{skc}^{(t)} \cdot \Delta t \cdot r_{sk}^{(t)} \tag{8}$$

IV. Joint Optimization Problem

With the aforementioned system and energy models, the problem of jointly maximizing the QoE of individual client i and minimizing the backhaul data traffic is formulated as the following integer non-linear programming (INLP) optimization model:

$$\underset{x,y,r,tr}{Maximize} \quad \alpha(\rho A Q_s - \omega E_s - \gamma F_s)\Delta t - (1-\alpha) BT_s \tag{9}$$

Subject to:

$$\sum_{s \in S} a_{sk}^{(t)} \cdot \lceil \frac{r_{sk}^{(t)}}{Thr_{sk}^{(t)}} \rceil \leq W_k^{(t)}, \quad \forall 1 \leq k \leq K, \;\; 1 \leq t \leq |T| \tag{10}$$

$$0 < L_s^{(t)} \leq L_s^{max}, \qquad \forall A_s \leq t \leq D_s \tag{11}$$

$$a_{sk}^{(t)} = \begin{cases} a_{sk}^{(t-1)}, & t \; mod \; C \neq 1 \\ 1, & t \; mod \; C = 1 \wedge k = arg \; \textbf{max}\{SNR_{sk}^{(t)}\} \\ 0, & \text{Otherwise} \end{cases} \tag{12}$$

$$x_{ske}^{(t)} + x_{skc}^{(t)} = 1, \quad \forall s \in S, \;\; A_s \leq t \leq D_s, \; 1 \leq k \leq K \tag{13}$$

$$\sum_{1 \leq k \leq K} y_{sk}^{(t)} \leq 1, \qquad \forall A_s \leq t \leq D_s \tag{14}$$

$$x_{skc}^{(t)}, x_{ske}^{(t)}, y_{sk}^{(t)} \in \{0,1\}, \quad \forall 1 \leq k \leq K, \;\; A_s \leq t \leq D_s \tag{15}$$

$$r_{sk}^{(t)}, tr_{sk}^{(t)} \in R, \qquad \forall 1 \leq k \leq K, A_s \leq t \leq D_s \tag{16}$$

In addition, the energy harvesting, consumption and the evolution models of the edge servers at each time slot as given in relations (1)-(4) are also included to the set of constraints in the above INLP problem. The only decision variables here are the binary variables $x_{ske}^{(t)}$, $x_{skc}^{(t)}$, $y_{sk}^{(t)}$ and the integer variables $r_{sk}^{(t)}$, $tr_{sk}^{(t)}$. Variables $L_s^{(t)}$, $b_k^{(t)}$ and $c_k^{(t)}$ are the dependent variables whose values depend on the values of the decision variables and, the remaining variables are independent and their values are known in advance.

Constraint (10) ensures that at each base station, the overall downlink resource blocks allocated to the associated clients at each time slot does not exceed the total available resource blocks at that time slot. It is noted that the ratio between the client's bitrate and its theoretical throughput at each time slot indicates the mapping resource blocks in the frequency domain at the base station at that time slot [23]. Constraint (11) ensures that no stalling happens in the client's buffer during its whole video streaming session and (12) determines the clients to server mapping. Constraint (13) ensures that the client downloads its chunk from only one location at each time slot and (14) states that at each time slot, the transrating operation can be performed in only one edge server. Constraints (15)-(16) determine the range of decision variables.

V. Proposed Online Algorithm

The existence of the integer decision variables in the optimization problem (9)-(16) makes it computationally intractable to solve using the exhaustive search methods. Further, the information about the clients is not known in advance which makes deployment of offline solutions practically unfeasible. To cope with these challenges, we design a heuristic-based online algorithm for the problem which takes advantage of in-network collaboration between the system entities. Our algorithm also utilizes a self-tuning mechanism which reduces the need for the parameterization of the optimization problem, thereby making it easy for the practical deployment. Pseudo-code of the proposed algorithm named energy-aware cache-based greedy bitrate allocation (ECGBA) is illustrated in Algorithm 1.

Algorithm 1: Energy-aware Cache-based Greedy Bitrate Allocation (ECGBA) Algorithm (Run by the Coordinators)

1: **Input:** $|T|, K, R$: Number of scheduling time slots, number of edge servers, set of available bitrates at origin server.
2: **Output:** Binary allocation $x_{skc}^{(t)}$, $x_{ske}^{(t)}$, $y_{sk}^{(t)}$ and integer bitrate allocation $r_{sk}^{(t)}$, $tr_{sk}^{(t)}$ for each client s, edge server $1 \leq k \leq K$ and time slot $1 \leq t \leq |T|$, $Utility$, $BackhaulTraffic$
3: **for** each time slot $1 \leq t \leq |T|$ **do**
4: **for** each edge server $1 \leq k \leq K$ **do**
5: $b_k^{(t)} = b_k^{(t)} + h_k^{(t)}$;
6: **for** each client s such that $A_s \leq t \leq D_s$ **do**
7: $maxUtility = -\infty$;
8: **if** $t = A_s$ **then**
9: **Initialize** $BufferStatus$ and $BT_s = 0$;
10: **Allocate** client s to server k according to (12)
11: **if** $(t - A_s)$ **mod** $C \neq 1$ **then**
12: **Allocate** client s to the same server and with same bitrate as with time slot $t-1$; **Update** $L_s^{(t)}$, BT_s;
13: **if** $BufferStatus = False$ **and** $L_s^{(t)} = L_s^{max}$ **then**
14: $BufferStatus = True$;
15: **if** $(t - A_s)$ mod $C = 1$ **then**
16: **Call** Subroutine Self-tuned Bitrate Selection;
17: **if** $t = D_s$ **then**
18: $Utility = Utility + maxUtility$;
19: $BackhaulTraffic = BackhaulTraffic + BT_s$;
20: **Return** $Utility$, $BackhaulTraffic$;

A. Energy-aware Cache-based Greedy Bitrate Allocation (ECGBA) Algorithm

At each time slot, the storage energy level of edge servers is first updated considering the amount of harvested renewable energy at that time slot (lines 4,5). For each active client at the current time slot, the algorithm then initializes its buffer status and the created backhaul traffic if the client starts its streaming session at the current time slot (lines 8,9). Then, the algorithm maps the client to the appropriate server using relation (12) such that the client is allocated to the same server,

with the same bitrate if it is downloading the middle of the chunk at the current time slot (lines 10-12). Otherwise, if the client is about to download the new chunk, the algorithm first assigns the client to the nearest edge server from where the client achieves the highest downlink SNR from the associated base station. The subroutine self-tuned bitrate selection is then called which allocates the most suitable bitrate at which the client downloads the new chunk of video (lines 15,16). Finally, if the client either leaves or finishes its streaming session at the current time slot, its utility (objective value (9)) and the corresponding backhaul traffic are added to the overall system outputs (lines 17-19).

B. Self-tuned Bitrate Selection

As part of the algorithm, the self-tuned bitrate selection procedure is executed if the client is about to download a new chunk and hence, the selection of most suitable bitrate for the new chunk should be decided. The pseudo-code of the procedure has been illustrated in Subroutine 1.

At first, the highest available bitrate is allocated to the client if it is downloading the first chunk of the video and its buffer level and the created traffic on the backhaul network are accordingly updated (lines 1-3). The estimated throughput and the switching threshold δ_{Sw}, which is used to control the switching level of the allocated bitrates to the consecutive chunks of the client, are then computed (line 4). Threshold δ_{Sw} is computed knowing that the highest switching between the consecutive chunks of video happens when the bitrates are allocated merely based on the buffer level [13]. A fairness threshold δ_{Fa} is also used to control the fairness value in allocating the bitrates to the client with respect to other simultaneous clients at the same time slot. Threshold δ_{Fa} is given as input to the algorithm at the deployment phase.

In a greedy manner, the utility objective value (9) is computed for all available bitrates (in decreasing order of magnitude) which are lower than the achievable throughput of the client (lines 5,6). Also, these bitrates should satisfy the resource allocation constraint at the base station (line 6) and both switching/fairness thresholds (line 7). The most suitable bitrate which has the maximum utility value is then chosen as the allocated bitrate to the current chunk of the client (lines 11-14). Note that the evaluation of function (9) for each bitrate is based on the availability of its corresponding video chunk in the local or neighborhood caches (line 8) or the possibility of transrating the bitrate at one edge server which holds the same chunk with higher bitrate and also has sufficient energy (storage or grid) to handle the transrating (lines 9,10).

If there is no such aforementioned bitrate available, the objective function is evaluated for those set of suitable bitrates which satisfy only the switching threshold compromising the fairness threshold (lines 15-19). Similarly, the candidate bitrate which maximizes the objective function (9) is chosen as the bitrate for the current chunk of the client. And, if none of the available bitrates satisfy even the switching threshold, the maximum suitable bitrate with highest achievable objective value is chosen as the bitrate for the current chunk of the client (lines 20-23). After the bitrate allocation to the current chunk, the weighting parameters of QoE term in (9) are dynamically computed (line 24).

Subroutine 1: Self-tuned Bitrate Selection

1: **if** $t - A_s \leq C$ **then**
2: **Allocate** the highest available bitrate;
3: **Update** *BufferStatus*, $L_s^{(t)}$, BT_s
4: **Compute** $estThr$ and threshold δ_{Sw};
5: **for** each bitrate $r \in R$ in decreasing order **do**
6: **if** $r \leq$ **max**$(estThr, \hat{Thr}_{sk}^{(t)}, L_s^{(t)})$ **and** allocation of r satisfy (10) **then**
7: **if** $|r - r_{sk}^{(t-1)}| \leq \delta_{Sw}$ **and** $1 - |r - \bar{r}|/(R_{max} - R_{min}) >= \delta_{Fa}$ **then**
8: **if** $(\lceil \frac{t-A_s}{C} \rceil, r) \in M_p^{(t)}, 1 \leq p \leq K$ **then** $Data = 0$
9: **else if** $\exists\ (k',\ tr > r) \ni (\lceil \frac{t-A_s}{C} \rceil, tr) \in M_{k'}^{(t)}$ **and** $(b_{k'}^{(t)}, grid\ energy\ budget\ k') - (tr - r)\phi \cdot e_p \geq 0$ **then** $Data = 0$;
10: **else** $Data = r$;
11: **Compute** weighting parameters ρ, ω and γ;
12: $QE = (\rho r - \omega|r - r_{sk}^{(t-1)}| - \gamma|r - \bar{r}|) \cdot \Delta t$;
13: **if** $\alpha QE - (1 - \alpha)Data > maxUtility$ **then**
14: $maxUtility = \alpha QoE - (1-\alpha)Data$; $r_{sk}^{(t)} = r$;
15: **if** $r_{sk}^{(t)} = 0$ **then**
16: **for** each bitrate $r \in R$ in decreasing order **do**
17: **if** $r \leq$ **max**$(estThr, \hat{Thr}_{sk}^{(t)}, L_s^{(t)})$ **and** allocation of r satisfy (10) **then**
18: **if** $|r - r_{sk}^{(t-1)}| \leq \delta_{Sw}$ **then**
19: **Perform** same operations as in lines 8-16;
20: **if** $r_{sk}^{(t)} = 0$ **then**
21: **for** each bitrate $r \in R$ in decreasing order **do**
22: **if** $r \leq$ **max**$(estThr, \hat{Thr}_{sk}^{(t)}, L_s^{(t)})$ **and** allocation of r satisfy (10) **then**
23: **Perform** same operations as in lines 8-16;
24: **Update** weighting parameters ρ, ω, γ at time slot t;
25: **Update** the binary decision variables $x_{skc}^{(t)}$, $x_{ske}^{(t)}$, $y_{sk}^{(t)}$ using the conditions in lines 7,8;
26: **Compute** AQ_s, E_s, F_s, BT_s and objective value of client s up to time slot t according to respectively (5), (6), (7), (8) and (9); **Update** $L_s^{(t)}$;
27: **if** $L_s^{(t)} = L_s^{max}$ **and** $BufferStatus = False$ **then**
28: $BufferStatus = True$;
29: **if** $y_{sk}^{(t)} = 1$ $(1 \leq k \leq K)$ **then**
30: $consumedEnergy = (tr - r_{sk}^{(t)}) \cdot \phi \cdot e_p$;
31: **if** $b_k^{(t)} - consumedEnergy \geq 0$ **then**
32: $b_k^{(t)} = b_k^{(t)} - consumedEnergy$;
33: **else**
34: $(grid\ energy\ k) = (grid\ energy\ k) - consumedEnergy$;
35: **Return** U_s, BT_s, $x_{skc}^{(t)}$, $x_{ske}^{(t)}$, $y_{sk}^{(t)}$;

The weighting of the average bitrate (ρ), bitrate switching (ω) and fairness (γ) at each time slot are computed based on how far the selected bitrate is from the optimal bitrate at that time slot. The procedure then proceeds with updating the values of decision variables (line 25) based on which location (the edge or the origin server) the allocated bitrate to the client has been either retrieved or processed. The created backhaul traffic and the objective value of the client along with the buffer status and buffer level are also accordingly updated (lines 26-28). Finally, if the allocated bitrate to the client has been processed at the edge, the corresponding energy

consumption of the grid or the energy level at the storage are accordingly updated (lines 29-34).

C. Retention-based Cache Replacement (RBC) Heuristic

After allocating the bitrates to all active streaming clients at each time slot, the edge servers in our system run independently a proactive cache replacement heuristic in order to update the cache contents if some of the chunks have been transferred from the origin server. We utilize a heuristic named retention-based cache replacement (RBC) which uses two sources of statistical information to make intelligent decisions about the eviction among multiple chunks for caching at each local edge server.

For each chunk/bitrate requested by a client, the heuristic first computes a caching value using two sources of statistical information: 1) How likely it is that the chunk/bitrate will be requested by the other clients in the future time slots and 2) how frequently the bitrate of that chunk has been requested by the clients (allocated to the same edge server) in the previous time slots. The information about the retention of the clients with respect to different video requests are used to approximately compute the first probability. We note that the origin server normally keeps the information about the clients' retention with respect to different videos which can be communicated with the edge servers. For instance, YouTube content delivery networks (CDNs) record some information about the clients' retention pattern when they watch some popular set of videos [28]. The second probability term which is in fact an estimation that the clients will request the bitrate in question according to their network conditions is computed based on their requested bitrates during the past time slots. The previous requested bitrates are readily available from the video streaming history of the clients. After computing the caching values for all chunks/bitrates at the current time slot, RBC heuristic then sorts the chunks in decreasing order of their caching values and inserts them into the cache until the cache is filled.

D. Computational Complexity

At each time slot, updating the storage energy of the edge servers take $O(K)$ in the worst case, where K is the number of edge servers. For each active client, the client to serve mapping task using relation (12) also takes $O(K)$ time. The most time-consuming part of the algorithm is then executing the self-tuned bitrate selection procedure which our analysis shows the worst case time complexity of $O(K+C\cdot S+|R|^2+|R|\cdot K+|T|)$ for this procedure. With overall S clients during $|T|$ time slots, the following worst-case time complexity is therefore obtained for ECGBA algorithm:

$$T_{ECGBA} \in O(|T| \cdot S \cdot K \cdot (T_{(Subroutine1)})) = O(|T| \cdot S \cdot K \cdot (K + C \cdot S + |R|^2 + |R| \cdot K + |T|)) \quad (17)$$

For the cache replacement heuristic RBC, we need to analyze the worst case time complexity at only one edge server since the servers run the heuristic independently. In the worst case, there is at least one new chunk downloaded from the origin server at each time slot and therefore, the edge server runs the heuristic for $|T|$ times in the worst case. At each time slot, our analysis including, the computation of caching likelihood for each requested chunk, sorting the chunks and inserting them into the cache, shows that the heuristic takes $O((Q/Cr_{min}+S)\cdot(S\cdot|T|+log(Q/Cr_{min}+S)))$ time. Q is the fixed cache size and r_{min} is the minimum available bitrate in set R. Therefore, with $|T|$ number of time slots, the following worst-case time complexity is obtained for RBC.

$$T_{RBC} \in O(|T| \cdot ((Q/Cr_{min}+S)\cdot(S\cdot|T| + log(Q/Cr_{min}+S)))) \quad (18)$$

VI. Simulation Results

In this section, we evaluate the performance of GMEC-enabled DASH system model through simulations using the radio access link level of mobile clients and emulated/measured solar energy harvesting patterns. Our main objectives are particularly to compare the following five strategies.

- **MEC-enabled Collaborative Edge Caching and Processing (CCP-MEC)**: Edge servers collaborate in caching and processing to serve the clients request and rely on the grid energy without the possibility of energy harvesting. The proposed RBC heuristic is also utilized for periodically updating the cache contents at the edge servers.
- **GMEC-enabled Collaborative Edge Caching and Processing (CCP-GMEC)**: Servers handle the clients request through collaborative caching and processing. They utilize the RBC heuristic and rely on both grid and periodically harvested renewable energy.
- **GMEC-enabled Non-collaborative Edge Caching and Processing (CP-GMEC)**: Edge servers serve the clients request from their own local caches independently and use the RBC heuristic for cache replacement. There is no collaboration with the neighborhoods and the servers rely on both grid and renewable energy for chunk transrating.
- **GMEC-enabled Collaborative Edge Caching and Processing using LRU Heuristic (CCP_LRU-GMEC)**: The adopted solution proposed in [7] in which the edge servers collaborate in caching and processing and have the potential of renewable energy harvesting. The common least recently used (LRU) heuristic is also used for edges cache replacement.

A. Simulation Setup

We consider a mobile edge computing scenario with 10 deployed edge servers (10 associated eNodeBs) and 100 mobile clients (UEs). The scheduling of the mobile clients is performed in one round which consists of $|T| = 300$ time slots where each time slot has the fixed duration of $\Delta t = 1$ $second$. The set of edge servers form one single cluster which means every edge server is in the neighborhood of the other servers. Under the urban macrocell wireless channel model specification [3GPP TR 36.814 V.9.0.0 2010] [24], the mobile clients (UEs) keep the constant speed of $4mps$ and their downlink SNR traces during 300 time slots are obtained from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2019.2918847, IEEE Transactions on Green Communications and Networking

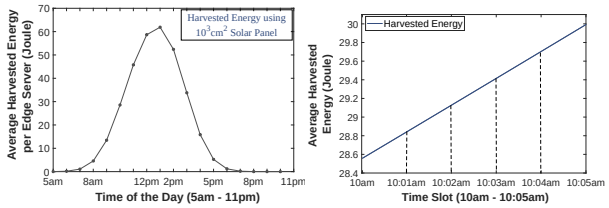
TABLE II: Simulation parameters and their values.

Simulation Parameter	Corresponding Value
Number of UEs	*100*
Number of eNodeBs	*10*
UE antenna gain	*0 dBi*
eNodeB antenna gain	*18 dBi*
UE speed	*4 mps*
Max Tx power per UE	*26 dBm*
Channel bandwidth	*5 MHz*
Shadowing	*Disabled*
Number of downlink RBs	*28*
Scheduler	*Proportional Fairness*
Channel model	*Urban Macrocell*
Simulation time	*300s*
Time slot duration	*1s*
Video chunk size	*5s*
Fairness threshold (δ_F)	*0.5*
Edge server cache size	*3Gb*
Processing weighting (ϕ)	*1*
Allocated grid energy to each edge server	*Unif[1.5KJ,2KJ]*
Energy Storage Capacity	*1KJ*
Max. Harvested Energy	*65J*
Energy Cons. per Bit (e_p)	$10^{-3}J$

the third party simulator SimuLTE [29]. The instantaneous effective throughputs of the clients are then obtained using the Shannon upper bound approximation.

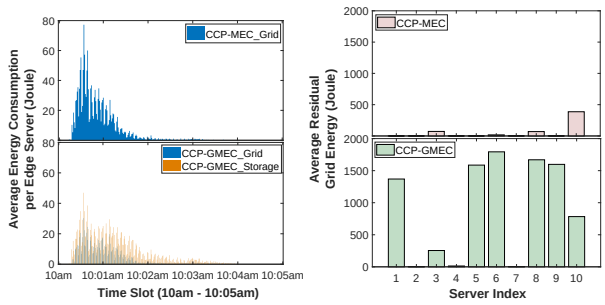
Four videos with different popularities are divided into consecutive chunks, each with fixed duration $C = 5s$ and are initially available at the origin server in ten different bitrates $R = \{15, 17, 22, 26, 30, 35, 38, 43, 45, 50Mbps\}$. Thanks to the availability of high bandwidth in upcoming 5G/B5G, these ranges of bitrates will be prevalent in next generation of mobile networks and have been considered in the related research studies [13], [30]. In order to generalize the results and consider varying network loads, different uniform intervals are considered for the time slot that the clients start the video streaming session (arrival). We further use the polynomial function $p(t) = at^2 + bt + c$ to model different retention behaviors of the clients with respect to multiple video requests where, $p(t)$ denotes the probability that the client will remain active in its video streaming session at time slot t. Different retention curves are generated by varying the curvature of polynomial $p(t)$ and determining the corresponding coefficients a, b and c.

Video buffer of the clients has also the fixed capacity of $L^{max} = 250Mb$. The size of the cache at each edge server is fixed at $Q_k = 3Gb, \forall 1 \leq k \leq K$ and the capacity of $B_e = 1KJoule$ is considered for the energy storage of edge servers. We adopt from [27] the Gaussian-shape solar energy harvesting profile with mean time slot chosen from the uniform interval $\mu \sim Unif[11am, 2pm]$ and the harvesting variance of $\sigma^2 = 5Joule$. According to [27], this energy harvesting profile corresponds to the maximum feasible harvesting (h_{max}) between approximately 60 to 80 $Joule$ at each time slot. We also assume that the available grid energy to each edge servers in one round of scheduling (300 seconds) is chosen from the uniform distribution $Unif[1.5KJ, 2KJ]$. Energy consumption of $e_p = 10^{-3}J$ is also considered for the processing of $1bit$ data at the edge server. The list of parameters setting for our simulations have been illustrated in Table. II. We also note that at each part, the average of the results taken over 20 runs of the simulation with confidence interval of 95% are presented.



(a) One day duration (b) Time interval 10-10:05am

Fig. 2: The pattern of harvested energy using 10^3cm^2 solar panel during (a) one day (b) time interval [10am,10:05am].



(a) Energy Consumption (b) Grid Energy Saving

Fig. 3: Comparison between CCP-MEC and CCP-GMEC in terms of (a) the average grid/storage energy consumption per edge server at each time slot and (b) grid energy saving.

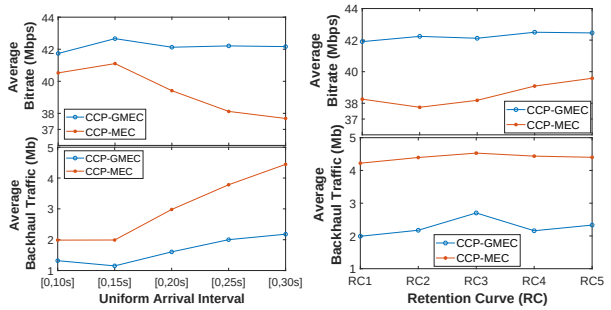
B. Solar Energy Harvesting Pattern

We have first plotted in Fig. 2a the pattern of average harvested solar energy per edge server during one day time duration from $5am$ until $11pm$ (excluding the mid-night intervals) using 10^3cm^2 solar panel. As it is seen, the peak harvested energy occurs during the noon-afternoon time periods with the maximum of approximately $h_{max} \approx 65Joule$ harvested energy. Since our system schedules the video streaming clients on one second time slot basis, we have considered the performance evaluation within 300 time slots of video streaming during the time interval $[10am, 10 : 05am]$. Fig. 2b, illustrates the pattern of harvested solar energy during this time interval which is utilized in our simulations.

C. Grid vs. Storage Energy Consumption

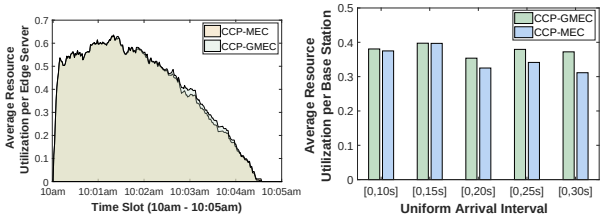
In Fig. 3a, we have compared two edge-enabled DASH solutions CCP-MEC and CCP-GMEC in terms of average grid and storage energy consumption per edge server during the video streaming interval ($5min$) considering the uniform arrival interval $Unif[0, 10s]$ and the linear retention curve. As it is observed, the integration of renewable harvested energy into the MEC indeed reduces on average about 50% the grid energy consumption per edge server. Fig. 3b shows that integrating the renewable energy also helps to save the grid energy about 90% for each edge server during the whole

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2019.2918847, IEEE Transactions on Green Communications and Networking



(a) Arrival Time (b) Retention Curve

Fig. 4: Comparison between collaborative caching and processing with MEC and GMEC in terms of average video bitrate and backhaul traffic per client time slot for different (a) arrival intervals and (b) retention curves.



(a) Arrival $Unif[0, 10s]$ (b) Different Arrivals

Fig. 5: Comparison between MEC and GMEC-enabled DASH in term of average resource utilization for (a) arrival interval $Unif[0, 10s]$ and (b) different arrival intervals.

video streaming session (scheduling round) compared to MEC without energy harvesting.

It is worthy pointing out that although the pattern of grid energy consumption in MEC and storage energy consumption in GMEC-enabled DASH are somewhat similar during the time interval $10am$-10:05am, the proposed GMEC-enabled DASH system model helps to save the grid energy for the time intervals when the intermittent renewable energy does not suffice for the edge processing. This in turn helps to significantly improve the processing efficiency of the servers during those time intervals and therefore improve the QoE of the mobile clients.

D. QoE and Backhaul Traffic Comparison

Next, we have compared two collaborative mobile edge caching and processing solutions in terms of average video bitrate and backhaul traffic per client time slot.

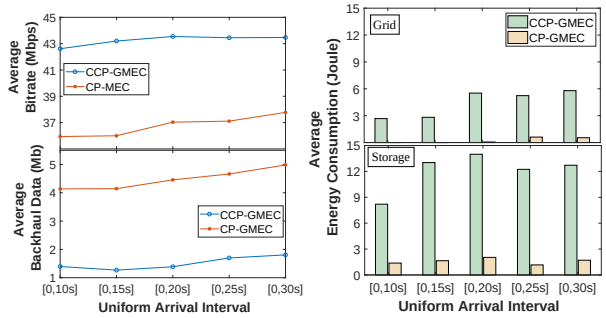
1) Video Bitrate: First, we set the coefficient of data traffic term in optimization problem to 1 ($\alpha = 0$) in order to evaluate the best possible improvement in average video bitrate using GMEC-enabled collaborative edge caching and processing (CCP-GMEC). Note that $\alpha = 0$ is the case that clients tend to fetch the chunks always from the local cache to minimize the backhaul traffic. Therefore, improving the processing capability of edge servers can indeed help to increase the average video bitrate of the clients. For different arrival intervals and retention curves, the comparison results have been illustrated in the top subplots in Fig. 4a and 4b. As the results show, using the storage renewable energy, the average video bitrate per client time slot increases. The reason is that with the help of storage energy, the processing capability of the edge servers increases which in turn yields downloading the chunks with higher bitrate from the local cache. As observed from the results, the average bitrate improvements of about 7% (for different arrival intervals) and 10% (different retention curves) are obtained using GMEC solution compared to MEC.

2) Backhaul Data Traffic: We have also compared two edge-enabled DASH solutions in term of average backhaul traffic per client time slot. For this simulation, we set the coefficient of QoE term in the optimization problem to $\alpha = 1$ in order to evaluate the best possible performance of CCP-GMEC in term of backhaul traffic reduction. Note that $\alpha = 1$ is the case that the clients tend to download the chunks always with highest possible bitrate regardless of the created backhaul traffic. Therefore, improving the processing capabilities of edge servers indeed helps to reduce the backhaul traffic. The results have been shown in the bottom subplots in Fig. 4a and 4b. As confirmed by the results, using the renewable energy can indeed help to reduce the volume of data traffic on the backhaul network. This is due to the fact that powering the servers with renewable energy when the energy of the internal battery suddenly drops, helps to improve the processing capability of the servers. This in turn causes the mobile clients to fetch some of their chunks/bitrates from the edges of the network rather than downloading from the origin server. As observed from the results, the average reductions of about 45% (for different arrival intervals) and 50% (for different retention curves) are achieved using GMEC solution.

It is noteworthy to mention that the improvements in QoE of the clients and noticeable backhaul traffic reduction by integrating the renewable energy into the MEC will result in increased mobile video streaming subscribers. This brings an overall revenue for MNO which will be significantly larger than the installation and maintenance costs of energy storages during long term system operation as it has been shown to be just about 2000$ for every 5 years [22].

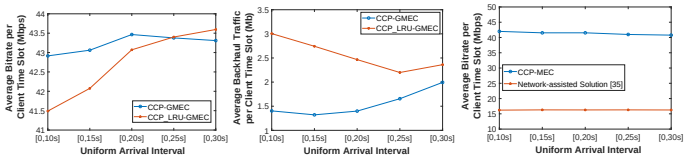
We have also shown the average resource utilization per base station (between zero and one) for both MEC and GMEC-enabled DASH solutions in Fig. 5a and Fig. 5b when $\alpha = 0$ in the joint optimization problem. With the clients arrival interval $Unif[0, 10s]$, Fig. 5a illustrates the average resource utilization per base station within the time interval $10am$-10:05am during the video streaming session. As wee see from the figure, the resource utilization is higher during the earlier times when the clients start their video streaming session and reduces over the time when the clients finish their streaming session. Furthermore, the results in Fig. 5b show that the change in resource utilization by varying the arrival intervals is not monotonic. Although the GMEC solution causes on average about 7% more resource utilization due to higher bitrates allocated to the clients, the wireless resources have not been however fully saturated.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2019.2918847, IEEE Transactions on Green Communications and Networking



(a) Bitrate/Backhaul Traffic (b) Energy Consumption

Fig. 6: Comparison between collaborative and non-collaborative edge caching and processing in terms of (a) average video bitrate/backhaul traffic (b) average grid/storage energy consumption.



(a) Average Bitrate (b) Average Traffic (c) Average Bitrate

Fig. 7: Comparison between our algorithm and another GMEC-enabled collaborative solution in terms of (a) average bitrate and (b) average backhaul traffic per client time slot (c) comparison to another network-assisted solution [31].

E. Comparison to Non-collaborative GMEC

Next, we have been interested to compare our solution with MEC integrated with renewable energy harvesting but without considering the collaboration among the edge servers i.e. the edge servers handle the clients independently from their own local caches. Since both approaches utilize the potential of energy harvesting, we set the weighing parameter $\alpha = 0.5$ in order to have a fair comparison. The comparison results in terms of average video bitrate and backhaul traffic for different arrival intervals of the clients have been shown in Fig. 6a.

As the results show, the collaborative caching and processing among the edge servers yields the average improvement of about 17% in video bitrate of the clients while the reduction of on average about 66% in the backhaul traffic. The reason is that with collaborative caching, the clients' request can be retrieved or processed from the neighborhood edge servers hence reducing the number of access to the origin server through the backhaul network. However, the non-collaborative caching saves on average about 94% and 85% the energy of respectively the grid and the storage of edge servers compared to collaborative GMEC as confirmed by the results in Fig. 6b. The reason is that in non-collaborative caching approach, the edge servers consume the energy of grid and storage for the processing of only the requests from the local clients while the collaborative solution consumes the energy for also the requests from the neighborhood clients.

F. Comparison to other Collaborative and Network-assisted Adaptation Solutions

We have also compared our solution with another collaborative edge caching and processing approach in MEC environments which is adopted from [7]. The bitrate adaptation part of the solution in this work is client-based and the common least recently used (LRU) heuristic is used to update periodically the cache contents at the edge servers. To have a fair comparison, we adopt this solution to use our network-assisted bitrate adaptation and further, the edge servers have the potential of renewable energy harvesting. The results of comparing our solution with the other collaborative approach in terms of average video bitrate and the backhaul data traffic for different arrival intervals of the clients have been shown in Fig. 7a and 7b.

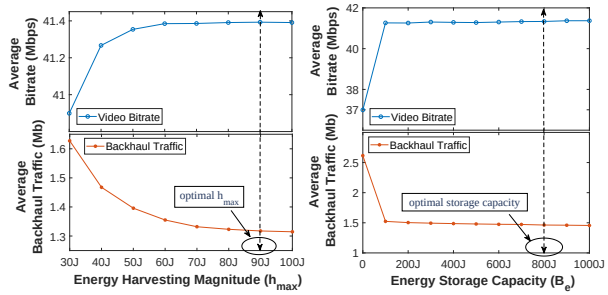
As we observe from the results, our algorithm improves the other solution marginally (about 1%) in term of average video bitrate while achieving significant improvement of about 40% in term of the average backhaul traffic per client time slot. The reason is that our solution gets advantage of some retention-based statics when updating the cache contents at the edge servers which in turn increases the cache hit rate and hence improving the performance. Although for few cases the other collaborative solution slightly improves the average bitrate compared to our algorithm, it however causes higher traffic on the backhaul network for those cases.

We have further compared our bitrate selection algorithm with another network-assisted solution which is adopted from [31]. The approach in this work solves a simple QoE utility objective function subject to the limited bandwidth on the bottleneck link which is shared among the set of competing clients. With the same number of clients and without energy harvesting, the comparison results in term of average video bitrate of the clients has been shown in Fig. 7c. As it is observed from the result, using our bitrate selection algorithm which uses the self-tuned parameterization technique results in the improvement of about 60% in term of average video bitrate per client time slot compared to the other solution.

G. Impact of Energy Harvesting Magnitude

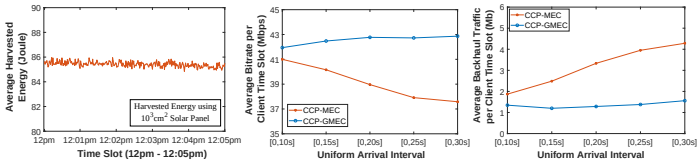
In the next part of simulation, we have been interested to investigate the impact of increasing the magnitude of energy harvesting on the system performance. For the purpose of this simulation, we have increased h_{max}, maximum feasible harvested energy at each time slot, from 30 $Joule$ to 100 $Joule$ while keeping the same storage capacity at $B_e = 1000 Joule$. Corresponding to each h_{max}, the average video bitrate ($\alpha = 0$) and backhaul data traffic ($\alpha = 1$) per client time slot have been illustrated in Fig. 8a. As the results show, higher available renewable energy for harvesting contributes to further improvement in the average bitrate of the clients as well as further reduction in the traffic on the backhaul network. The reason is that the available energy at the edge servers is actually the limiting constraint to achieve higher video bitrates and therefore, the possibility of harvesting more energy alleviates this constraint.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TGCN.2019.2918847, IEEE Transactions on Green Communications and Networking



(a) Harvested Energy (b) Energy Storage Capacity

Fig. 8: Impact of increasing the (a) magnitude of energy harvesting and (b) the capacity of energy storage.



(a) Measured Energy (b) Video Bitrate (c) Backhaul Traffic

Fig. 9: (a) Measured energy harvesting between $[12pm - 12 : 05pm]$ (adopted from [32]) and the comparison results in terms of (b) average video bitrate (c) average backhaul traffic.

As we see from the results, the achievable bitrate of the clients and the corresponding backhaul traffic reach a stable level after a threshold on the energy harvesting magnitude. The reason is that although more energy harvesting helps to further improve the average video bitrate of the clients, however, the limited cache size at the edges and the weighting $\alpha = 0$ in the joint optimization problem limit the maximum achievable bitrate. Similarly, the weighting factor $\alpha = 1$ limits the minimum backhaul traffic that can be achieved. As observed from Fig. 8a, the optimal energy harvesting magnitude of $h_{max} = 90Joule$ was obtained for this simulation.

H. Impact of Energy Storage Capacity

We have also investigated the impact of increasing the capacity of the energy storage on the system performance in terms of average video bitrate and backhaul traffic. For this simulation, we have increased the size of storage from $B_e = 0$ to $B_e = 1000Joule$ while keeping the same magnitude of energy harvesting at $h_{max} = 65Joule$. Corresponding to each storage size, the average bitrate and backhaul traffic per client time slot have been plotted in Fig. 8b. As expected, the average bitrate and backhaul traffic both improve when with fixed energy harvesting magnitude, the size of the energy storage increases. However, no further improvement can be achieved after a threshold point. The reason is that although there are enough space for storing the harvested energy with large energy storage size, however, the system performance is upper bounded by the amount of harvested energy. As observed from the result in Fig. 8b, the threshold point of $B_e = 800Joule$ is noticed in our simulation.

I. Performance Evaluation using Measured Solar Radiation

We have also evaluated the performance of GMEC-enabled DASH system model using the real measured solar energy harvesting pattern which has been extracted from the experimental results reported in [32]. In this work, the peak solar power generated during the month of January in Hamburg city of Germany is reported approximately $85mWh/cm^2$. This energy harvesting corresponds to about $10^3 \times 85mWh = 85Wh$ using $10^3 cm^2$ solar panel. The pattern of harvested energy during $5min$ time duration within the interval $[12pm, 12 : 05pm]$ with $\pm 1W$ radiation deviation has been shown in Fig. 9a.

Using the downlink SNR of the mobile clients and the measured energy harvesting pattern given in Fig. 9a, the results of comparing GMEC-enabled DASH with MEC without energy harvesting have been illustrated in Fig. 9b and Fig. 9c. As the results show, the proposed GMEC-enabled DASH system achieves on average about 8% and 57% improvements in terms of respectively average video bitrate and average backhaul traffic per client time slot. These results confirm the superiority of GMEC-enabled DASH system using real energy harvesting measurement.

VII. Conclusion and Future Work

This paper investigates the impact of integrating the renewable energy into the edge computing known as green mobile edge computing (GMEC) on the joint optimization of QoE of the mobile clients and the backhaul traffic in particularly dynamic adaptive video streaming over HTTP (DASH) scenarios. Due to the NP-hardness of the formulated joint optimization problem, we design a low-complexity greedy-based algorithm using a self-tuning parametrization technique to solve the problem. Results of our performance evaluations using downlink SNR of mobile clients and with both simulated and measured energy harvesting patterns reveal that the integration of renewable energy into the MEC indeed helps to decrease the grid power consumption of the edge servers, reduce significantly the backhaul data traffic and improve the average video bitrate of the clients.

In this work, we assumed that edge servers handle the processing requests of the connected or the neighborhood clients by relying on their available grid and renewable energy. It is expected that considering the cooperation among neighborhood edge servers in energy sharing known as geographically load balancing (GLB) concept brings further improvements in system performance. Furthermore, the utilization of caching at mobile devices and D2D communication among the neighborhood clients can result in further reducing the grid energy consumption of edge servers as well as further alleviating the traffic burden on backhaul network. We consider these directions as our future research works.

Acknowledgment

This research has been financially supported by Lacrimosa project grant number 297892 and the Nokia Center for Advanced Research.

REFERENCES

[1] I. Cisco, "Cisco visual networking index: Forecast and methodology, 2016-2021", CISCO White paper, Sep. 2017.

[2] ISO/IEC 23009-1. 2014. Dynamic adaptive streaming over HTTP (DASH) Part 1: Media presentation description and segment formats. (2014)

[3] T-Y. Huang, R. Johari, N. McKeown, M. Trunnell, and M. Watson "A buffer-based approach to rate adaptation: Evidence from a large video streaming service", *in Proc. ACM Conf. on SIGCOMM (SIGCOMM'14)*, pp. 187-198, Aug. 2014.

[4] T. Mangla, N. Theera-Ampornpunt, M. Ammar, E. Zegura, and S. Bagchi, "Video through a crystal ball: Effect of bandwidth prediction quality on adaptive streaming in mobile environments", *in Proc. 8th ACM Int. Workshop on Mobile Video*, pp. 1-6, May 2016.

[5] M. Patel, B. Naughton, C. Chan, N. Sprecher, S. Abeta, A. Neal, "Mobile-edge computing", ETSI Introductory Technical White Paper, Sep. 2014.

[6] D. Liu, B. Chen, C. Yang, and A. F. Molisch, "Caching at the wireless edge: Design aspects, challenges and future directions", *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 22-28, Sep. 2016.

[7] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks", *in Proc. 13th Annual IEEE Conf. on Wireless On-demand network systems and services (WONS)*, pp. 165-172, Feb. 2017.

[8] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "QoE-traffic optimization through collaborative edge caching in adaptive mobile video streaming", *IEEE Access*, vol. 6, pp. 52261-52276, Sep. 2018.

[9] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices", *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590-3605, Dec. 2016.

[10] J. Xu, and S. Ren, "Online learning for offloading and autoscaling in renewable-powered mobile edge computing", *in Proc. IEEE Global Commun. Conf. (Globecom).*, pp. 1-6, Dec. 2016.

[11] J. Kua, G. Armitage and P. Branch, "A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP", *IEEE Commun. Surveys & Tuts.*, vol. 19, no. 3, pp. 1842-1866, Third Quarter 2017.

[12] G. Cofano, L. D. Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo, "Design and experimental evaluation of network-assisted strategies for HTTP adaptive streaming", *in Proc. 7th ACM Int. Conf. on Multimedia Systems (MMSys'16)*, pp. 1-12, May 2016.

[13] A. Mehrabi, M. Siekkinen, and A. Ylä-Jääski, "Edge computing assisted adaptive mobile video streaming", *IEEE Trans. Mobile Comput.*, pp. 1-17, Jun. 2018, doi: 10.1109/TMC.2018.2850026.

[14] X. Wang, M. Chen, T. Taleb, A. Ksentini, V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems", *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 131-139, Feb. 2014.

[15] X. Li, P. Wu, X. Wang, K. Li, Z. Han and V. C. M. Leung, "Collaborative hierarchical caching in cloud radio access networks", *in Proc. IEEE Conf. on Comput. Commun. Workshops (INFOCOM)*, pp. 462-467, May 2017.

[16] H. A. Pedersen and S. Dey, "Enhancing mobile video capacity and quality using rate adaptation, RAN caching and processing", *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 996-1010, Apr. 2016.

[17] T. X. Tran, A. Hajisami, P. Pandey and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges", *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54-61, Apr. 2017.

[18] M. H. Rehmani, M. Reisslein, A. Rachedi, M. Erol-Kantarci, and M. Radenkovic, "Integrating renewable energy resources into the smart grid: Recent developments in information and communication technologies", *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 2814-2825, Jul. 2018.

[19] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys & Tuts*, vol. 19, no. 4, pp. 2322-2358, Fourth Quarter 2017.

[20] M. Mendil, A. D. Domenico, V. Heiries, R. caire, and N. Hadjsaid, "Battery-aware optimization of green small cells: Sizing and energy management", *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 3, pp. 635-651, Sep. 2018.

[21] P-H. Chiang, R. B. Guruprasad, and S. Dey, "Optimal use of harvested solar, hybrid storage and base station resources for green cellular networks", *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 3, pp. 707-720, Sep. 2018.

[22] X. Liu, N. Ansari, "Dual-Battery enabled profit driven user association in green heterogeneous cellular networks", *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 4, pp. 1002-1011, Dec. 2018.

[23] J. Chen, R. Mahindra, M. A. Khojastepour, S. Rangarajan, and M. Chiang, "A scheduling framework for adaptive video delivery over cellular networks", *in Proc. ACM Annual Int. Conf. Mobile Comput. Netw. (MobiCom'13)*, pp. 389-400, Sep. 2013.

[24] LTE Guideline: http://www.etsi.org/deliver/etsi_tr/136900_136999/136942/08.02.00_60/tr_136942v080200p.pdf

[25] 3GPP Specifications: https://www.etsi.org/deliver/etsi_ts/136200_136299/136213/13.00.00_60/ts_136213v130000p.pdf

[26] M. Zhao, X. Gong, J. Liang, W. Wang, X. Que, and S. Cheng, "QoE-driven cross-layer optimization for wireless dynamic adaptive streaming of scalable videos over HTTP", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 3, pp. 451-465, Mar. 2015.

[27] T. D, Nguyen, J. Y. Khan, and D. T. Ngo, "A distributed energy-harvesting-aware routing algorithm for heterogeneous IoT networks", *IEEE Trans. Green Commun. Netw.*, vol. 2, no. 4, pp. 1115-1127, Dec. 2018.

[28] M. Siekkinen, M. A. Hoque, and J. K. Nurminen, "Using viewing statics to control energy and traffic overhead in mobile video streaming", *IEEE/ACM Trans. Netw.*, vol. 24, no. 3, pp. 1489-1503, Jun. 2016.

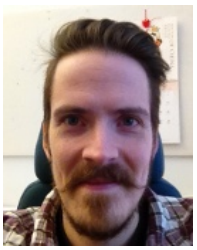
[29] SimuLTE: http://www.simulte.com

[30] C. Ge, N. Wang, G. Foster, and M. Wilson, "Toward QoE-assured 4K video-on-demand delivery through mobile edge virtualization with adaptive prefetching", *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2222-2237, Oct. 2017.

[31] Z. Li, S. Zhao, D. Medhi, and I. Bouazizi, "Wireless video traffic bottleneck coordination with a DASH SAND framework", *in Proc. IEEE Vis. Commun. Image Process.*, pp. 1-4, Nov. 2016.

[32] D. Krüger, S. Fischer, and C. Buschmann, "Solar power harvesting - Modeling and experiences", *GI/ITG KuVS Fachgespräch Drahtlose Sensornetze*, vol. 8, 2009.

Abbas Mehrabi received the BSc degree in computer engineering from the Shahid Bahonar University of Kerman, Iran, in 2008, the MSc degree in computer engineering from Azad University, South Tehran, in 2010, and the PhD degree from the School of Electrical Engineering and Computer Science, Gwangju Institute of Science and Technology, Gwangju, South Korea, in 2017. He is currently a postdoctoral researcher with the Department of Computer Science, Aalto University, Espoo, Finland. His main research interests include quality of experience optimization and resource allocation in mobile edge computing environments, Internet of Things, vehicular fog computing, and scheduling/planning problems in smart grids.



Matti Siekkinen received the M.Sc. degree in computer science from the Helsinki University of Technology in 2003, and the Ph.D. degree from the EURECOM/University of Nice Sophia-Antipolis in 2006. He is currently a Senior Researcher with the University of Helsinki and Aalto University. His research on multimedia systems combines techniques from multimedia signal processing, mobile networking, cloud computing, system analysis, machine learning, and HCI.

Antti Ylä-Jääski received the Ph.D. degree from ETH Zurich in 1993. From 1994 to 2009, he was with Nokia in several research and research management positions, with a focus on future Internet, mobile networks, applications, services, and service architectures. Since 2004, he has been a tenured Professor with the Department of Computer Science, Aalto University. His current research interests include mobile cloud computing, mobile multimedia systems, pervasive computing and communications, indoor positioning and navigation, energy efficient communications and computing, and Internet of Things.